

seRmS

sehr einfache Registermaschinen-Simulation von Ferdinand Winhard unter Verwendung der JGUI-Toolbox von Hans Witt

Für die Nutzung der Simulation muss auf dem Rechner ein Java Runtime Environment installiert sein. Die aktuelle Version erhält man zum Beispiel unter <http://www.java.com/de/download/manual.jsp>. Ansonsten ist keinerlei Installation erforderlich. Die Simulation wird mit Doppelklick auf das Executable Jar File seRmS gestartet.

Der dazu gehörige Assembler umfasst folgende Befehle:

Befehl	Auswirkung
DLOAD i	lädt unmittelbar die Zahl i in A, erhöht den Wert in BZ um 1
LOAD x	kopiert den Wert in Rx nach A, erhöht den Wert in BZ um 1
STORE x	kopiert den Wert in A nach Rx, erhöht den Wert in BZ um 1
ADD x	addiert den Wert in Rx zum Wert in A, legt das Ergebnis in A ab, erhöht den Wert in BZ um 1
SUB x	subtrahiert den Wert in Rx vom Wert in A, legt das Ergebnis in A ab, erhöht den Wert in BZ um 1
MULT x	multipliziert den Wert in Rx mit dem Wert in A, legt das Ergebnis in A ab, erhöht den Wert in BZ um 1
DIV x	dividiert den Wert in A durch den Wert in Rx (Ganzzahldivision ohne Rest), legt das Ergebnis in A ab, erhöht den Wert in BZ um 1
JUMP n	lädt die Zahl n in BZ, Programm wird mit Befehl in Speicherzelle n fortgesetzt (unbedingter Sprung)
JGE n	lädt die Zahl n in BZ, falls der Wert in A größer oder gleich 0 ist, erhöht ansonsten den Wert in BZ um 1 (bedingter Sprung)
JGT n JLE n JLT n JEQ n JNE n	entsprechende bedingte Sprünge für die Fälle "größer als 0" "kleiner oder gleich 0" "kleiner als 0" "gleich 0" "ungleich 0"
END	erhöht den Wert in BZ um 1 und beendet den Programmlauf

Das Programm wird mit einem Texteditor erstellt und in einer Datei gespeichert. Dabei ist zu beachten, dass das Programm in Zelle 1 beginnt und die weiteren Befehle in fortlaufend nummerierten Zellen liegen. Jede Programmzeile beginnt mit der Zellnummer gefolgt von einem Doppelpunkt. Nach einem oder mehreren Leerzeichen folgt der Befehl. Nach einem Trennzeichen (üblicherweise --) können Kommentare eingefügt werden. Es stehen 16 Register (R0 bis R15) zur Verfügung. Ein gültiges Programm endet immer mit dem Befehl END.

Zum Laden des Programms wird, falls sich Simulation und Programmdatei im selben Verzeichnis befinden, der Dateiname, andernfalls der vollständige Pfad (im UNIX-Format mit '/') in das Eingabefenster eingetragen. Werden Programmdateien in einem Unterverzeichnis des Verzeichnisses gespeichert, in dem sich die Simulation befindet, kann der Pfad auch mit diesem Unterverzeichnis beginnen. In jedem Fall startet ENTER das Einlesen. Beim Dateinamen ist zu beachten, dass viele Editoren ein Suffix .txt automatisch anhängen, dieses aber im Dateimanager oft nicht angezeigt wird.

Ein Beispielprogramm:

Register	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15
A	5															
BZ			8													
R0	0	336	6	1												
R1					0	0	0	0	0	0	0	0	0	0	0	0
R2						0	0	0	0	0	0	0	0	0	0	0
R3							0	0	0	0	0	0	0	0	0	0
R4								0	0	0	0	0	0	0	0	0
R5									0	0	0	0	0	0	0	0
R6										0	0	0	0	0	0	0
R7											0	0	0	0	0	0
R8											0	0	0	0	0	0
R9											0	0	0	0	0	0
R10											0	0	0	0	0	0
R11											0	0	0	0	0	0
R12											0	0	0	0	0	0
R13											0	0	0	0	0	0
R14											0	0	0	0	0	0
R15											0	0	0	0	0	0

Anfang **Schritt** **Durchlauf**

Mithilfe dreier Tasten kann das Programm schrittweise oder ganz abgearbeitet und die Maschine auf den Anfangszustand zurückgesetzt werden. Dabei wird jeweils der nächste abzuarbeitende Befehl farbig hervorgehoben.

Disclaimer

Die Simulation wurde für die Verwendung im Informatikunterricht entwickelt. Der Autor übernimmt keinerlei Haftung für Schäden, die durch die Verwendung der Software mittelbar oder unmittelbar entstehen. Auch kann keinerlei Support geleistet werden.

Für Anregungen und konstruktive Kritik bin ich stets dankbar.

Juni 2010
Ferdinand Winhard
winhard@vr-web.de